NORTHEASTERN UNIVERSITY

EECE 5554

ROBOTICS SENSING AND NAVIGATION

FINAL PROJECT REPORT

on

Visual SLAM using ORB SLAM-3

GROUP 3

Instructor: Thomas Consi

Name	**NUID**
Yutika Chougule	NUID:002767787
Amey Kore	NUID:002108884
Rucha Pendharkar	NUID:002783508
Pratik Baldota	NUID:002761711
Rohit Kothawale	NUID:002724449

Introduction

We will be implementing a Visual SLAM Algorithm for path planning using a monocular camera. The algorithm will be tested on EuRoC and KITTI Datasets. To confirm the effectiveness of the implemented algorithm, we will be developing a python driver for collecting and plotting GPS coordinates using ROS. The GPS data will be used for comparison. Parallelly, the algorithm will be implemented on GPS and Camera Data collected by the NUANCE car. It is accurate, precise and versatile. It will compute real time trajectories and 3D reconstructions.

SLAM is an essential piece in robotics that helps robots to estimate their pose – the position and orientation – on the map while creating a map of the environment to carry out autonomous activities.

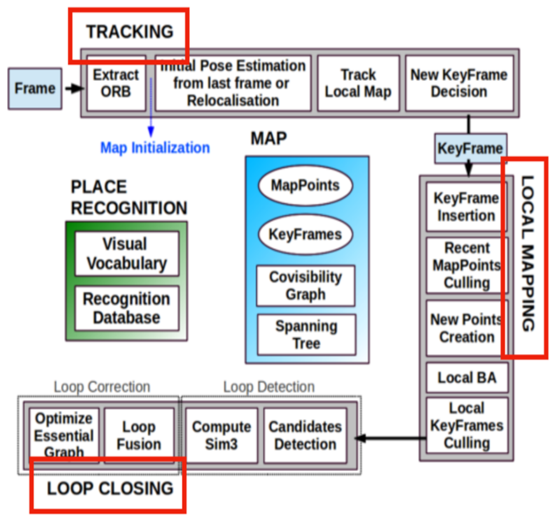

Fig 1: System Architecture of ORB-SLAM3

The above figure represents the system architecture of ORB-SLAM3, and it has the following structures:

- **Tracking Thread:** It extracts ORB features and computes pose in the frame and locates each frame from Atlas maps.
- **Atlas and DBoW2 Keyframe Database:** It stores unique keyframes and contains a localized active map.
- **Local Mapping Thread:** It Adds keyframes to the active map, removes duplicate keyframes, performs bundle adjustment, and refines IMU parameters with map estimation.
- **Loop & Map Merging Thread:** It Identifies common regions in the active map, stores keyframes, merges different maps, and performs loop correction.
- **Full Bundle Adjustment:** It updates the map for reprojection errors.

Methodology

In this project we utilized the official GitHub repository of ORB-SLAM3 from the authors: Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, Juan D. Tardos.

- ORB-SLAM3 was implemented on three datasets – EuRoC, KITTI and the NUANCE Car datasets.
- We performed Monocular and Stereo ORB-SLAM3 on the morning_stereo_rgb_ir_lidar_gps.bag dataset that was collected from Nuance Car.
- We subscribed to the topic /camera array/cam0/camera info to obtain camera data
- Nuance.yaml files were modified and used to perform ORB-SLAM3 on monocular camera data
- ORBvoc.txt is essentially a database of previously seen feature descriptors stored in a formatted file for fast place recognition to trigger loop closure instead of matching features one by one against all images. This is a generic file that is already trained and could be replaced with a similar file from custom training.
- Nuance.yaml includes parameters like camera calibration and distortion parameters, camera width, camera height, camera frames per second, ORB extraction feature, etc.
- We chose 6000 feature points to perform ORB- SLAM3 on Monocular data. This was decided after tweaking the value and performing SLAM to see the best result possible.
- Similarly, ORB-SLAM3 was applied on Stereo camera data to see if it performed better than monocular.

Results on NUANCE dataset

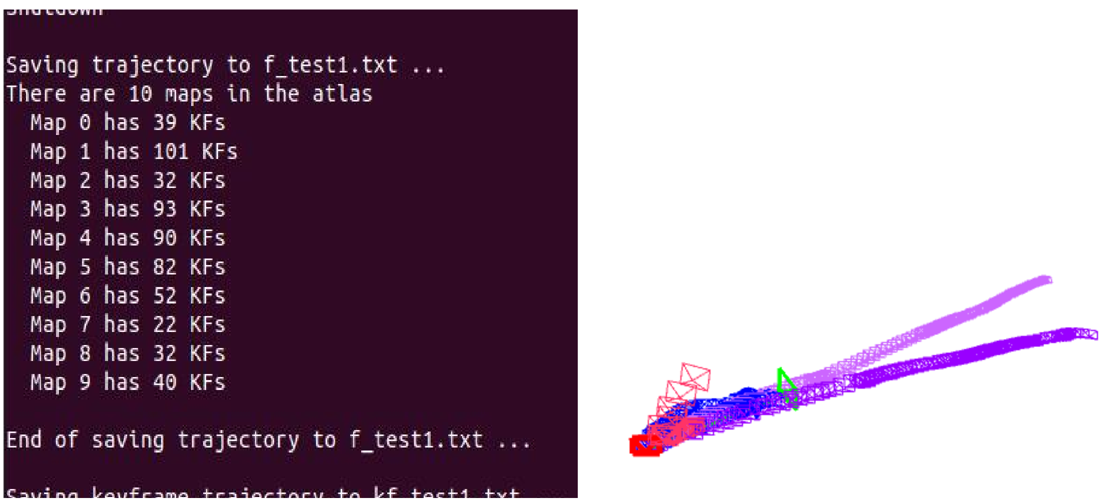

Fig.2. Multiple Trajectories generated in Monocular Nuance

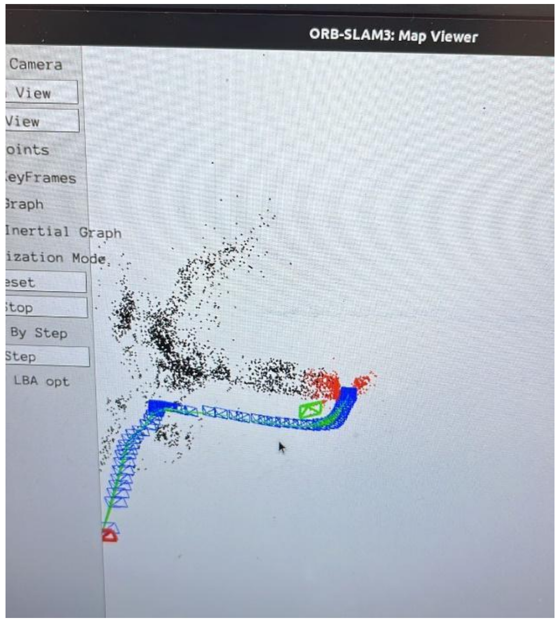

Fig.3 Two turns detected on monocular camera dataset

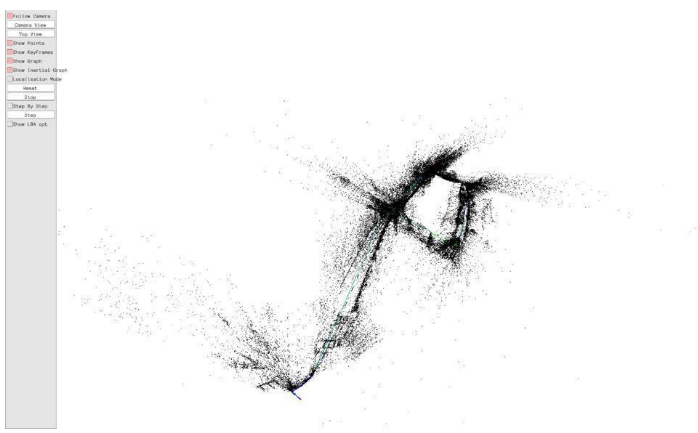

Fig.4 Loop Closure on Nuance Dataset using ORB-SLAM using Stereo Camera

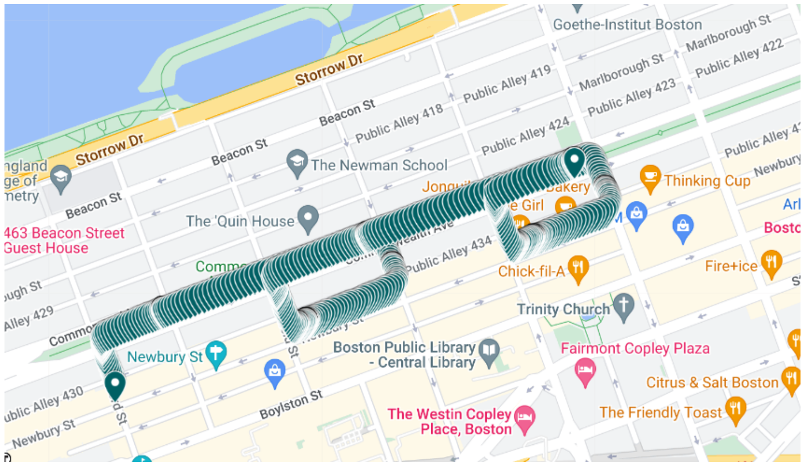

Fig.5 NUANCE GPS plot

Observations

- For monocular ORB-SLAM3, we can observe from the above results that the NUANCE dataset fails when the car takes steep turns. The features are not being detected; hence tracking does not take place. This can be because images might be a little distorted. (Fig.2)
- This results in many small maps rather than one map and means when we stitch them together at the end to estimate our overall trajectory, it basically looks like a straight line, since none of the turns are successfully included. As there were 8 turns, we can observe 9 maps have been generated (i.e., new map at every turn). (Fig.2)
- We tried tweaking the parameters for monocular camera data, but we only managed to detect 2 turns. (Fig.3.)
- On the other hand, ORB-SLAM3 on Stereo data got loop closures and thus better results (Fig.4). Similar pattern is followed when compared to the GPS plot (Fig.5).
- ORB-SLAM yielded better results for EuRoC, KITTI followed by NUANCE on stereo data.

Conclusion

- The ORB-SLAM algorithm produces better results for slow turns and for indoor environments as opposed to outdoor. It does not perform well during fast turns
- Stereo Cameras are more efficient than monocular cameras for implementing ORB- SLAM in outdoor environments.
- To overcome the limitation of Monocular ORB-SLAM, sensor fusion methods such as Monocular-Inertial SLAM, which uses IMU data, can be used.

References

- https://github.com/UZ-SLAMLab/ORB_SLAM3
- https://arxiv.org/pdf/2007.11898.pdf%5C
- https://ouster.com/blog/introduction-to-slam-simultaneous-localization-and-mapping/